From Text to Talk: Audio-Language Model Needs Non-Autoregressive Joint Training

Anonymous authors
Paper under double-blind review

Abstract

Recent advances in large language models (LLMs) have attracted significant interest in extending their capabilities to multimodal scenarios, particularly for speech-to-speech conversational systems. However, existing multimodal models handling interleaved audio and text rely on autoregressive (AR) methods, overlooking that text depends on target-target relations whereas audio depends mainly on source-target relations. In this work, we propose Text-to-Talk (TtT), a unified audio-text framework that integrates AR text generation with non-autoregressive (NAR) audio diffusion in a single Transformer. By leveraging the any-order AR property of absorbing discrete diffusion, our approach provides a unified training objective for text and audio. To support this hybrid generation paradigm, we design a modality-aware attention mechanism that enforces causal decoding for text while allowing bidirectional modeling within audio spans, and further introduce three training strategies that reduce train-test discrepancies. During inference, TtT employs block-wise diffusion to synthesize audio in parallel while flexibly handling variable-length outputs. Comprehensive experiments on Audio-QA, ASR, AAC and speech-to-speech benchmarks show that TtT consistently surpasses strong AR and NAR baselines, with additional ablation and training-strategy analyses confirming the contribution of each component. We will open-source our models, data and code to facilitate future research in this direction.

1 Introduction

The recent success of LLMs has catalyzed a paradigm shift towards general-purpose Multimodal Large Language Models (MLLMs) capable of processing and generating information across diverse modalities (Xu et al., 2025; Team et al., 2023). Among these, speech-to-speech conversational systems have emerged as a pivotal component in facilitating natural human-AI interaction. Conventional systems typically decompose this problem into a cascaded pipeline of Automatic Speech Recognition (ASR), LLM-driven response generation, and Text-To-Speech (TTS) synthesis. While effective to a degree, this modular design introduces significant latency accumulation and error propagation between modules, hindering naturalness and real-world applicability. In response, recent end-to-end approaches like Moshi (Défossez et al., 2024), GLM4-Voice (Zeng et al., 2024), and VITA-Audio (Long et al., 2025) have sought to unify speech understanding and generation within a single model. These models are typically trained through multi-stage pipelines that involve text-to-audio tokenizer training, interleaved data construction, text-audio alignment and task-oriented supervised fine-tuning (Huang et al., 2025; Li et al., 2025; Ding et al., 2025; Chu et al., 2024). As shown in Figure 1, these methods aim to generate interleaved text and speech tokens in an autoregressive (AR) manner, which are then decoded into continuous audio waveforms by a separate neural codec or diffusion-based decoder (Mehta et al., 2024; Kong et al., 2020).

However, this emerging paradigm faces a fundamental challenge. As illustrated in Figure 1, we identify a fundamental mismatch in prevailing approaches that employ a single language model to autoregressively generate both text and audio tokens (Zeng et al., 2024; Xie & Wu, 2024b; Borsos et al., 2023; Dang et al., 2024; Rubenstein et al., 2023). This uniform treatment applies identical AR training objectives across both modalities, overlooks a critical distinction in their underlying generative processes. Text generation inherently follows a sequential causal structure characterized by strong **target-target** dependencies (Box et al., 2015), where each token explicitly conditions on previously generated tokens. Consequently, an incorrect token prediction can propagate and intro-

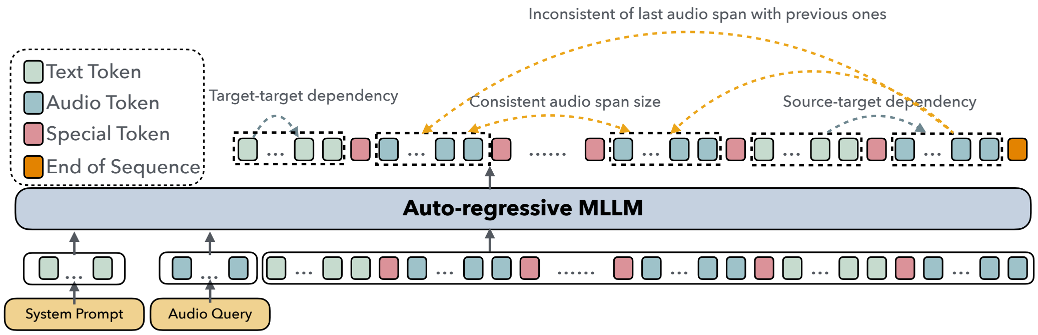


Figure 1: (a) Distinct dependency structures for text and audio modality. (b) Due to disparate tokenization rates, the last audio span is of variable length.

duce subsequent errors due to the exposure bias inherent in AR models (Ranzato et al., 2015). In contrast, audio token generation is predominantly driven by **source-target** dependencies (Ren et al., 2020), where audio output primarily condition on the source text rather than on the preceding audio tokens. Specifically, within the current non-autoregressive (NAR) span, audio tokens generation should remain faithful to the source text even when previous audio tokens are incorrectly predicted. Applying a purely AR objective to audio generation thus introduces unnecessary sequential constraints, leading to suboptimal training dynamics and magnifying error propagation. This problem can be substantially alleviated by adopting a NAR generation strategy, which aligns better with the source-dependent nature of audio modeling. Recently, discrete diffusion has emerged as a compelling alternative to AR for discrete sequence modeling (Yu et al., 2025; Gong et al., 2024; Austin et al., 2021; Sahoo et al., 2024). Beyond empirical gains, recent theory shows that absorbing discrete diffusion can be interpreted as modeling the conditional distributions of clean tokens and admits a tight connection to any-order AR objectives (Ou et al., 2024).

Thus, we introduce Text-to-Talk (TtT), a unified audio-text MLLM that integrates AR text generation with NAR audio diffusion within a single Transformer initialized from a pretrained LLM. Text segments are trained with a standard AR cross-entropy objective, while audio segments are modeled via an NAR discrete diffusion process. During inference, the model dynamically switches between AR and NAR decoding strategies based on special control tokens. In summary, our work makes the following contributions:

- We identify and formalize the fundamental asymmetry in dependency structures between text and audio modalities—text exhibits target-target dependencies requiring causal ordering, while audio is driven by source-target dependencies. Leveraging the any-order AR nature of absorbing discrete diffusion, we establish a unified theoretical framework that proves our joint training objective provides an upper bound on the negative log-likelihood of the desired joint distribution.
- We propose TtT, a hybrid AR-NAR MLLM that seamlessly integrates AR text generation with discrete diffusion-based audio synthesis within a single Transformer initialized from a pretrained LLM. Our design preserves the reasoning and instruction-following capabilities of the base LLM while enabling efficient parallel audio generation.
- We introduce three principled training strategies to address the inherent train-test discrepancies in hybrid AR-NAR learning, enabling stable training and robust content-aware variable-length generation that bridges the gap between training and inference conditions.
- Extensive experiments across Audio-QA, ASR, AAC and speech-to-speech benchmark demonstrate that TtT consistently outperforms strong AR and NAR baselines, highlighting the advantage of the hybrid AR–NAR framework.

2 Preliminary and Notation

In this section, we establish the basic notation for interleaved audio-text sequences and provide brief overviews of the two core generative paradigms employed in our framework: AR modeling and absorbing discrete diffusion. These form the theoretical foundation of our proposed method in Section 3.

Tokens, Vocabulary, and Interleaved layout We consider interleaved discrete text–audio sequences of length L: $x = (x^1, \ldots, x^L)$ with a unified discrete vocabulary $\mathcal{V} = \mathcal{V}_{\text{text}} \cup \mathcal{V}_{\text{audio}} \cup \mathcal{S}$, where $\mathcal{S}$ contains special tokens such as $\langle\text{SOA}\rangle$ (start of audio), $\langle\text{EOA}\rangle$ (end of audio), $\langle\text{EOS}\rangle$ (end of sequence) and the absorbing mask token $[\mathbf{M}]$. A sequence x is structured as a series of alternating text and audio spans: $x = (\mathcal{T}_1, \mathcal{A}_1, \ldots, \mathcal{T}_M, \mathcal{A}_M, \langle\text{EOS}\rangle)$, where:

- $\mathcal{T}_m = (t_{m,1}, \ldots, t_{m,|\mathcal{T}_m|}) \in (\mathcal{V}_{\text{text}} \cup \{\langle\text{EOS}\rangle, \langle\text{SOA}\rangle\})^{|\mathcal{T}_m|}$ are text tokens
- $\mathcal{A}_m = (a_{m,1}, \ldots, a_{m,|\mathcal{A}_m|}) \in (\mathcal{V}_{\text{audio}} \cup \{\langle\text{EOA}\rangle\})^{|\mathcal{A}_m|}$ are quantized audio tokens

Let $f_\theta : \mathcal{V}^L \to \mathbb{R}^{L \times d}$ be a single Transformer (e.g. Qwen 2.5). We use a shared output head $W \in \mathbb{R}^{d \times |\mathcal{V}|}$ (typically tied with input embeddings) to produce per-position logits over the entire vocabulary $\mathcal{V}$.

AR Modeling AR models factorize the joint probability of a sequence $x = (x^1, \ldots, x^L)$ into a product of conditional probabilities, based on the chain rule: $p(x) = \prod_{i=1}^{L} p(x^i | x^{<i})$, where $x^{<i} = (x^1, \ldots, x^{i-1})$. This imposes a sequential, causal structure on the generation process. For a detailed discussion, please refer to Appendix A.3.1.

Absorbing Discrete Diffusion Absorbing discrete diffusion models are a NAR paradigm for sequence generation. They consist of a forward process that corrupts a clean sequence by gradually replacing tokens with a special absorbing mask state $[\mathbf{M}]$, and a learned reverse process that aims to recover the original sequence from the corrupted input. A key insight from Ou et al. (2024) is that the learning objective simplifies to modeling a time-independent conditional probability of the clean data. Specifically, the score for unmasking a token v at a corrupted position is given by:

$$\underbrace{\frac{p_t(\ldots, \hat{x}^i{=}v, \ldots)}{p_t(\ldots, x^i = [\mathbf{M}], \ldots)}}_{\text{concrete score}} = \underbrace{\frac{e^{-\bar{\sigma}(t)}}{1 - e^{-\bar{\sigma}(t)}}}_{\text{time scalar}} \cdot \underbrace{p_0\big(v \mid UM\big)}_{\text{clean conditional probability}} . \tag{1}$$

where UM denotes the set of unmasked (visible) tokens and t represents the continuous time step of the corruption process. This denoising formulation is precisely the objective of an Any-Order AR Model (AO-ARM), predicting a token given an arbitrary context of unmasked tokens. As demonstrated by Ou et al. (2024), the diffusion training objective is mathematically equivalent to this AO-ARM objective, which averages the prediction loss over all possible permutations of the sequence: $\mathcal{L}_{AO}(\boldsymbol{x}_0) = \mathbb{E}_{\pi \sim U_\pi} \sum_{l=1}^{d} - \log q_\theta(x_0^{\pi(l)} | x_0^{\pi(<l)})$, where π is a random permutation of the token indices. Therefore, training an absorbing discrete diffusion model is equivalent to training a powerful ensemble of AR models that can operate in any order. More details in Appendix A.3.2.

3 Joint Text-AR & Audio-NAR Model

In this section, we introduce our proposed model, integrates AR generation for text and discrete diffusion for audio within a single, unified Transformer architecture.

3.1 AR Modeling for Text

We model text generation using a fixed, canonical auto-regressive order. Let π_{text} denote the natural left-to-right permutation over all text token positions in the sequence — that is, $\pi_{\text{text}}(1) < \pi_{\text{text}}(2) < \cdots < \pi_{\text{text}}(|\mathcal{T}_{\leq M}|)$ where $\mathcal{T}_{\leq M} = \cup_{m=1}^{M} \mathcal{T}_m$ is the set of all text token indices.

At the span level, the probability of generating the m-th text span $\mathcal{T}_m = (t_{m,1}, \ldots, t_{m,|\mathcal{T}_m|})$ conditioned on all prior context is given by: $p_\theta(\mathcal{T}_m | \mathcal{T}_{<m}, \mathcal{A}_{<m}) = \prod_{j=1}^{|\mathcal{T}_m|} p_\theta(t_{m,j} | \mathcal{T}_{<m}, \mathcal{A}_{<m}, t_{m,<j})$, where $t_{m,<j} = (t_{m,1}, \ldots, t_{m,j-1})$ is the prefix of text tokens within the current text span.

To express the joint probability of all text tokens in the sequence, we account for the conditioning on preceding audio spans. The joint probability $p_\theta(x_{text})$ is therefore defined as the product of the probabilities of each text span, conditioned on all prior spans: $p_\theta(x_{text}) = \prod_{m=1}^{M} p_\theta(\mathcal{T}_m | \mathcal{T}_{<m}, \mathcal{A}_{<m}) = \prod_{m=1}^{M} \prod_{j=1}^{|\mathcal{T}_m|} p_\theta(t_{m,j} | \mathcal{T}_{<m}, \mathcal{A}_{<m}, t_{m,<j})$,

The model is trained by minimizing the standard causal cross-entropy loss over all text positions:

$$\mathcal{L}_{\text{AR}}(x) = -\sum_{m=1}^{M}\sum_{j=1}^{|\mathcal{T}_m|}\log p_\theta\big(t_{m,j} \mid \mathcal{T}_{<m}, \mathcal{A}_{<m}, t_{m,<j}\big) \tag{2}$$

3.2 Absorbing discrete diffusion for audio spans

Building on the theoretical foundation established in Section 2, we apply absorbing discrete diffusion to audio spans $\mathcal{A}_{\leq M} = \cup_{m=1}^{M}\mathcal{A}_m$. This design choice aligns with the fundamental difference in dependency structures: audio tokens exhibit strong **source→target** dependencies (conditioning on source text), making them well-suited for the any-order AR nature of diffusion, while text tokens follow **target→target** causal dependencies, better handled by standard AR modeling.

Audio-specific Corruption and Denoising For each training sample, we sample a masking level $\lambda \sim U([0,1])$ and independently mask each audio token with probability λ, while preserving all text tokens. This creates corrupted sequences where audio spans contain a mixture of original tokens and mask tokens $[\mathbf{M}]$, but text spans remain intact. To enable efficient parallel training across all audio spans simultaneously, we apply masking operations to every audio span $\mathcal{A}_m$ in the sequence, rather than processing them sequentially. This parallel masking strategy significantly improves training efficiency while leveraging the time-independent nature of the denoising objective (Eq. 1).

Training Objective for Audio Generation The model learns to predict the original audio tokens for masked positions by minimizing the λ-denoising cross-entropy loss over all audio spans. As discussed in (Ou et al., 2024), this objective is mathematically equivalent to the any-order AR objective, and can be equivalently expressed in the AO-ARM form:

$$\mathcal{L}_{\text{AO}}(x) = \sum_{m=1}^{M}\mathbb{E}_{\pi_m\sim U_{\pi_m}}\sum_{j=1}^{|\mathcal{A}_m|} -\log q_\theta(a_{m,\pi_m(j)}|\mathcal{T}_{\leq m}, \mathcal{A}_{<m}, a_{m,\pi_m(<j)}) \tag{3}$$

where π_m is a random permutation over the positions within audio span $\mathcal{A}_m$, and $a_{m,\pi_m(<j)}$ denotes the audio tokens that appear before position j in the permuted order. This formulation makes explicit that the audio generation objective is learning to predict each audio token conditioned on an arbitrary subset of other tokens within the same span, plus the full cross-modal context from text. This any-order AR nature is what enables parallel generation during inference.

3.3 Multimodal Factorization and Unified Objective

Having established AR modeling for text in Section 3.1 and discrete diffusion for audio in Section 3.2, we now formalize how these two paradigms can be unified within a single probabilistic framework. The key insight is to leverage the distinct dependency structures of each modality through a *partial-order factorization* that respects the causal nature of text while allowing flexible ordering within audio spans. Recall that text tokens exhibit strong target-target dependencies requiring causal ordering, while audio tokens primarily depend on source-target relationships with their corresponding text. This suggests that within each audio span $\mathcal{A}_m$, the tokens can be generated in any order as long as they condition on the appropriate cross-modal context $\mathcal{T}_{\leq m} \cup \mathcal{A}_{<m}$. We formalize this intuition using partial orders over token positions.

A partial order on a set V is a binary relation $\preceq$ that is reflexive, antisymmetric, and transitive. A set equipped with such a relation is called a partially ordered set (poset). Two elements $a, b \in V$ are comparable if $a \preceq b$ or $b \preceq a$; otherwise, they are incomparable. An antichain is a subset of V in which every pair of distinct elements is incomparable — that is, no internal ordering constraints exist among them (Davey & Priestley, 2002).

Partial-order Formulation Let $(V, \preceq)$ be a poset over all token indices in the sequence, where V represents all token positions and $\preceq$ encodes precedence relationships. For our interleaved text-audio setting, we define: (1) Each text token $t_{m,j}$ precedes $t_{m,j+1}$ (maintaining left-to-right causality within text spans). (2) All tokens in span m precede all tokens in span $m+1$ (maintaining cross-span dependencies). (3) Tokens within each audio span $\mathcal{A}_m$ form an antichain under $\preceq$ (no *mandatory* internal ordering), but the model is permitted to condition on previously generated tokens within the same span during training and inference under any linear extension.

For any token i, let $\mathrm{Pa}(i)$ denote its set of predecessors under this partial order. By construction, each audio token $a_{m,j}$ has predecessors $\mathrm{Pa}(a_{m,j}) = \mathcal{T}_{\leq m} \cup \mathcal{A}_{<m}$, while for text tokens $\mathrm{Pa}(t_{m,j}) = \mathcal{T}_{<m} \cup \mathcal{A}_{<m} \cup t_{m,<j}$.

Any linear extension ℓ of the partial order $(V, \preceq)$ induces a valid chain-rule factorization: $p(x) = \prod_{j=1}^{|V|} p(x_{\ell(j)} \mid x_{\mathrm{Pa}(\ell(j))})$. Since audio tokens within each span form an antichain, there are multiple valid linear extensions differing only in the within-span ordering of audio tokens. Rather than committing to a single extension, we can *marginalize* over all possible orderings within audio spans.

Order-marginalized Factorization for Audio Spans For an antichain $S \subseteq V$ (such as tokens within an audio span), we define the *order-marginalized conditional* by averaging over all permutations of S: $\tilde{p}_\theta(x_S \mid x_{V\setminus S}) = \mathbb{E}_{\pi \in \mathrm{Perm}(S)} \prod_{j\in S} q_\theta(x_{\pi(j)} \mid x_{V\setminus S}, x_{\pi(<j)})$, where $q_\theta(\cdot \mid \cdot)$ represents the any-order AR learned through discrete diffusion. When applied to our audio spans, this gives:

$$\tilde{p}_\theta(\mathcal{A}_m \mid \mathcal{T}_{\leq m}, \mathcal{A}_{<m}) = \mathbb{E}_{\pi_m \sim U_{\pi_m}} \prod_{j=1}^{|\mathcal{A}_m|} q_\theta\big(a_{m,\pi_m(j)} \,\big|\, \mathcal{T}_{\leq m}, \mathcal{A}_{<m}, a_{m,\pi_m(<j)}\big) \tag{4}$$

Intuitively, this averages the likelihood over all possible within-span orderings, reflecting the fact that audio tokens can be generated in any order given the appropriate cross-modal context. Note that while tokens within $\mathcal{A}_m$ form an antichain under the partial order (i.e., no mandatory sequential constraints), the order-marginalized conditional in Eq. 4 allows the model to leverage local target-target dependencies that may arise under specific generation orders. This flexibility enables the model to capture useful intra-span structures when beneficial.

Hybrid AR-NAR Joint Distribution Combining fixed-order AR for text with order-marginalized factorization for audio, our model induces the joint scoring function:

$$\tilde{p}_\theta(x) = \prod_{m=1}^{M} \Bigg[\underbrace{\prod_{j=1}^{|\mathcal{T}_m|} p_\theta\big(t_{m,j} \,\big|\, \mathcal{T}_{<m}, \mathcal{A}_{<m}, t_{m,<j}\big) \cdot}_{\text{single-order AR for text}} \quad \underbrace{\tilde{p}_\theta(\mathcal{A}_m \mid \mathcal{T}_{\leq m}, \mathcal{A}_{<m})}_{\text{order-marginalized any-order AR for audio}} \Bigg] \tag{5}$$

This formulation reveals that both modalities are fundamentally AR: text uses a single linear extension (left-to-right), while audio integrates over all linear extensions consistent with the partial order.

Training Objective and Upper Bound Analysis In practice, we cannot directly optimize $\tilde{p}_\theta(x)$ because the order-marginalized conditional in Eq. 4 requires computing expectations over all permutations. Instead, we use the training objectives $\mathcal{L}_{\text{AR}}(x)$ and $\mathcal{L}_{\text{AO}}(x)$ derived in Section 3.2. The key theoretical insight is that our combined training objective provides a tight upper bound on the negative log-likelihood of the desired joint distribution. To see this, consider the audio term:

$$\begin{aligned} &\mathbb{E}_{\pi_m \sim U_{\pi_m}} \sum_{j=1}^{|\mathcal{A}_m|} \big[-\log q_\theta\big(a_{m,\pi_m(j)} \mid \mathcal{T}_{\leq m}, \mathcal{A}_{<m}, a_{m,\pi_m(<j)}\big)\big] \\ &\qquad \geq -\log \mathbb{E}_{\pi_m \sim U_{\pi_m}} \prod_{j=1}^{|\mathcal{A}_m|} q_\theta\big(a_{m,\pi_m(j)} \mid \mathcal{T}_{\leq m}, \mathcal{A}_{<m}, a_{m,\pi_m(<j)}\big) \end{aligned} \tag{6}$$

The right-hand side is precisely $-\log \tilde{p}_\theta(\mathcal{A}_m \mid \mathcal{T}_{\leq m}, \mathcal{A}_{<m})$ from Eq. 4. The left-hand side is exactly the audio loss term for span m in our practical training objective $\mathcal{L}_{\text{AO}}(x)$.

To establish the unified upper bound, we now sum the inequality in Eq. 6 over all audio spans $m = 1, \ldots, M$:

$$\sum_{m=1}^{M} \mathbb{E}_{\pi_m} \sum_{j=1}^{|\mathcal{A}_m|} \left[-\log q_\theta\big(a_{m,\pi_m(j)} \mid \mathcal{T}_{\le m}, \mathcal{A}_{<m}, a_{m,\pi_m(<j)}\big)\right] \ge \sum_{m=1}^{M} \big(-\log \tilde{p}_\theta(\mathcal{A}_m \mid \mathcal{T}_{\le m}, \mathcal{A}_{<m})\big) \tag{7}$$

The left-hand side is exactly $\mathcal{L}_{\text{AO}}$. For the text terms, the $\mathcal{L}_{\text{AR}}$ loss is defined in Eq. 2. Combining the text and audio terms according to the joint factorization in Eq. 5 yields:

$$\mathcal{L}_{\text{Unified}}(x) \triangleq \mathcal{L}_{\text{AR}}(x) + \mathcal{L}_{\text{AO}}(x) \ge -\log \tilde{p}_\theta(x) \tag{8}$$

a detailed derivation of this inequality is provided in Appendix A.1.1, where this final inequality follows from combining the text equality with the audio inequality derived above. Thus, minimizing our practical training objective $\mathcal{L}_{\text{Unified}}(x)$ corresponds to minimizing an upper bound on the negative log-likelihood of the theoretically motivated joint distribution $\tilde{p}_\theta(x)$. This result is significant because: (1) It provides theoretical justification for our hybrid AR-NAR training approach. (2) It guarantees that optimizing the computationally tractable objective $\mathcal{L}_{\text{Unified}}(x)$ will not deviate arbitrarily from the theoretically optimal objective $-\log \tilde{p}_\theta(x)$.

Training Pipeline and Loss Computation Our training pipeline starts from a pretrained text LLM and expands its vocabulary with discrete audio codebook tokens and control symbols ($\langle$SOA$\rangle$, $\langle$EOA$\rangle$). Each training sequence is organized as interleaved text spans and audio spans. We provide an illustration of loss computation in Appendix A.5. Despite its theoretical and practical advantages, the hybrid AR-NAR paradigm introduces a significant train-test discrepancies that can degrade generation quality. During training, audio spans are partially masked according to the diffusion process, while during inference, the model must generate audio and text tokens conditioned on complete text context and previously generated clean audio tokens. To bridge this gap, we propose three principled training strategies:

- **Batchwise AR & NAR Objective Mixing (BANOM)**: With probability p_{mix}, we skip the diffusion noise addition process for certain samples and compute gradients only on text tokens using AR loss. This ensures that during training, text tokens occasionally observe clean, unmasked audio spans—matching the inference scenario where text generation conditions on previously generated complete audio content rather than partially masked spans.
- **Prefix Preservation Masking (PPM)**: For a fraction p_{prefix} of training samples, we randomly select a cutoff index m and ensure that all preceding audio spans $\mathcal{A}_{<m} = \{\mathcal{A}_1, \ldots, \mathcal{A}_{m-1}\}$ remain unmasked, while applying NAR diffusion loss only to spans $\mathcal{A}_{\ge m} = \{\mathcal{A}_m, \mathcal{A}_{m+1}, \ldots, \mathcal{A}_M\}$. This strategy ensures that during training, when generating span $\mathcal{A}_m$, the model observes clean representations of all previous spans $\mathcal{A}_{<m}$, matching the inference scenario where audio spans are generated sequentially and each span $\mathcal{A}_m$ conditions on fully generated, clean preceding spans $\mathcal{A}_{<m}$ rather than their corrupted, partially masked versions.
- **Stochastic Span Truncation (SST)**: We address the positional bias in $\langle$EOA$\rangle$ prediction by randomly truncating audio span $\mathcal{A}_M$ during training. Due to disparate tokenization rates between text and audio, audio tokens significantly outnumber text tokens, resulting in fixed-size spans $\mathcal{A}_1, \ldots, \mathcal{A}_{M-1}$ and a variable-length final span $\mathcal{A}_M$. Since all audio spans undergo simultaneous diffusion training, the model learns to predict $\langle$EOA$\rangle$ at fixed positions for early spans, creating a strong positional bias that hinders content-aware termination learning for the final span. To mitigate this, we implement stochastic truncation: with probability p_{trunc}, we randomly select a truncation length $k < |\mathcal{A}_M|$ and create a truncated span $\mathcal{A}_M^{\text{trunc}} = (a_{M,1}, \ldots, a_{M,k})$ by removing the original $\langle$EOA$\rangle$ token and suffix tokens $(a_{M,k+1}, \ldots, a_{M,|\mathcal{A}_M|})$. This creates training samples where span termination occurs at arbitrary positions rather than fixed boundaries, forcing the model to predict $\langle$EOA$\rangle$ based on semantic content and contextual text rather than positional cues.

3.4 Modality-Aware Attention Mechanism

Our attention design enforces a step-wise pattern across three content types: (1) the input prompt uses standard causal attention; (2) text tokens $\mathcal{T}_m$ apply strict causal attention to the prompt, all

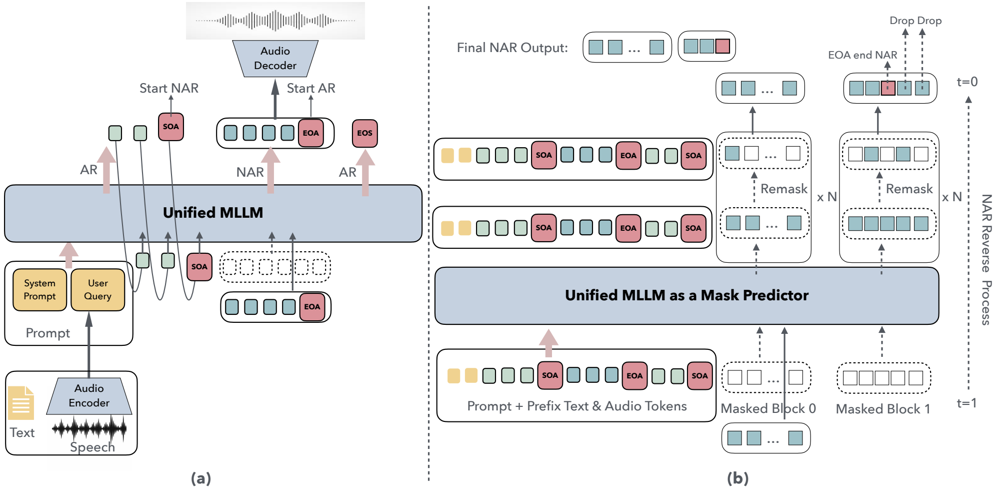


Figure 2: **Overview of the proposed framework and its diffusion reverse process. (a) TtT framework.** A unified MLLM that interleaves AR text and NAR audio generation. The model alternates between AR text decoding and NAR audio synthesis based on control tokens. **(b) Diffusion reverse process.** NAR audio generation through iterative denoising

prior spans, and preceding tokens within their span; and (3) audio tokens $\mathcal{A}_m$ use hybrid attention—bidirectional within their own span and causal attention to the prompt and all earlier spans. This pattern enables parallel audio span training in a single forward pass while preventing cross-span interference. See Appendix A.5 for an illustration.

3.5 Inference Process

Figure 2 shows the overview of TtT and its inference process. During inference, TtT alternates between AR text decoding and NAR audio synthesis within a unified framework. Given input audio, the encoder produces discrete tokens, which are processed by AR generation until $\langle$SOA$\rangle$ is reached. The model then switches to NAR mode, where block-wise diffusion (see Algorithm 1 in Appendix A.4) generates audio spans in parallel. Upon predicting $\langle$EOA$\rangle$, remaining tokens in the block are dropped and decoding returns to AR mode, repeating the cycle until $\langle$EOS$\rangle$ is generated. Each completed audio span is immediately sent to the audio decoder, enabling parallel synthesis with low first-token latency and continuous streaming generation.

4 Experiments

4.1 Experimental Setups

Datasets To effectively train and evaluate our proposed TtT framework, we follow prior works (Zeng et al., 2025; Ding et al., 2025; Long et al., 2025) and adopt a diverse collection of multi-task datasets, including ASR, TTS, audio chat, text chat, Automated Audio Captioning (AAC), Speech Emotion Classification (SEC), Acoustic Scene Classification (ASC), and interleaved text–audio data. In total, the combined training corpus contains approximately 6.3 million samples across these tasks. For evaluation, we focus on three representative capabilities: (1) conversational reasoning via Audio-QA, (2) cross-modal alignment via ASR, and (3) audio comprehension via AAC. To further validate end-to-end speech-to-speech performance in realistic conversational scenarios, we additionally evaluate on URO-Bench (Yan et al., 2025), a comprehensive benchmark that integrates reasoning, understanding, and oral conversation tasks. Full details on training data composition and evaluation protocol are provided in Appendix A.6 and Appendix A.7.

Evaluation To effectively evaluate our model on these tasks, we carefully design the evaluation protocol and metrics: (1) For Audio-QA, we introduce an ASR-LLM pipeline that transcribes the model's spoken responses using language-specific ASR systems (Paraformer-zh for Chinese and Whisper-Large-v3 for English) and leverages a powerful LLM-as-a-Judge (Qwen3-235B-A30B)

Table 1: Comprehensive evaluation of TtT framework. Higher (↑) is better for Audio-QA, lower (↓) is better for ASR. Datasets abbreviations are available in Table 7

Models	**Audio-QA (↑)**				**ASR (↓)**					
	AE.	**LQ.**	**TQA.**	**WQ.**	**Fzh.**	**A2.**	**A1.**	**WS_m.**	**WS_n.**	**Fen.**
Main Results										
Qwen2.5-1.5B (AR)	**17.99**	16.78	1.61	2.32	99.08	59.73	80.27	85.55	81.76	96.16
Qwen2.5-1.5B (NAR)	10.70	0.00	0.40	0.20	86.97	224.37	191.11	123.96	143.76	108.25
TtT-1.5B (AR–NAR)	15.68	**23.75**	**3.47**	**7.70**	**44.36**	**14.89**	**16.72**	**52.23**	**41.52**	**49.00**
Qwen2.5-3B (AR)	14.42	10.00	0.60	0.70	90.32	54.94	72.01	80.01	73.64	74.47
Qwen2.5-3B (NAR)	11.31	0.67	1.21	0.70	68.94	212.27	160.58	89.22	111.29	83.51
TtT-3B (AR–NAR)	**17.46**	**34.68**	**6.53**	**11.61**	**55.67**	**12.53**	**13.65**	**53.83**	**44.29**	**64.31**
Ablation Study										
TtT-3B w/o BANOM	13.87	19.87	2.81	5.12	58.25	18.58	21.35	58.48	49.52	68.90
TtT-3B w/o PPM	14.27	22.79	2.71	5.54	58.86	15.63	18.83	57.76	47.92	67.37
TtT-3B w/o SST	14.12	10.20	1.30	3.72	56.39	25.43	31.03	64.41	56.70	62.60
TtT-3B (AR–NAR)	**17.46**	**34.68**	**6.53**	**11.61**	**55.67**	**12.53**	**13.65**	**53.83**	**44.29**	**64.31**
Training Strategy Comparison										
TtT-3B (AR–NAR)	17.46	34.68	6.53	11.61	55.67	12.53	13.65	53.83	44.29	64.31
Pretrain+AR	**29.45**	15.93	3.61	11.45	23.37	9.79	12.67	**26.75**	20.91	19.49
Pretrain+TtT	26.73	**40.07**	**11.07**	**21.43**	**18.99**	**6.80**	**5.78**	27.59	**19.85**	**19.10**

to assess semantic correctness against ground-truth answers; (2) For the ASR task, we directly measure transcription accuracy using Word Error Rate (WER); (3) For the AAC task, we adopt the evaluation prompt from CLAIR-A (Wu et al., 2024), using thinking model Qwen3-30B-A3B to judge the caption quality; (4) For URO-Bench, we directly use the official evaluation code and protocol to ensure a fair comparison with existing systems; More details are in Appendix A.7.1.

Baselines We compare TtT with state-of-the-art audio-language models, including Moshi Défossez et al. (2024), SpeechGPT Zhang et al. (2023), Kimi-Audio Ding et al. (2025), VITA-Audio Long et al. (2025), LLaMA-Omni Fang et al. (2025), GLM-4-Voice Zeng et al. (2024), Mini-Omni Xie & Wu (2024b) and SLAM-Omni Chen et al. (2025) (detailed descriptions in Appendix A.8).

Model Configuration We adopt the Qwen2.5-Base model as the backbone, experimenting with parameter scales of 1.5B and 3B, and fine-tune all parameters during training. For the audio components, we directly follow the audio tokenizer and decoder design introduced in GLM-4-Voice (Zeng et al., 2024). These modules have been shown to provide efficient and high-quality speech tokenization and synthesis, and they allow our framework to leverage strong audio modeling without requiring additional architectural modifications. The training details are provide in Appendix A.9.

4.2 Validating the Hybrid AR-NAR Architecture

To evaluate the effectiveness of our proposed TtT framework, we compare it with two representative variants, purely AR backbone and purely diffusion based NAR backbone. For fairness and scalability, all three frameworks are instantiated with backbones of 1.5B and 3B parameters.

Performance Analysis on Audio-QA and ASR Tasks Table 1 (Main Results) provides the comparative results for the Audio-QA and ASR tasks. Our proposed TtT framework consistently outperforms both pure AR and NAR variants across all metrics. Specifically, at the 3B scale, TtT-3B surpassing Qwen2.5-3B (AR) by +3.04, +24.68, +5.93, and +10.91. For ASR tasks, TtT-3B yielding improvements of 42.41 and 58.36 absolute WER points over Qwen2.5-3B (AR). These substantial gains stem from our hybrid AR–NAR design: the NAR diffusion component enables efficient parallel denoising for tighter audio–text alignment, capturing audio's inherent source-target dependencies, while AR text generation maintains coherent cross-modal conditioning and respects target-target dependencies. In contrast, purely NAR models perform notably worse due to order confusion from applying order-agnostic objectives to inherently sequential text-audio sequences. We also observe consistent scaling trends, where TtT-3B substantially outperforms TtT-1.5B across all tasks.

Table 2: Performance comparison on Audio-QA, ASR, and AAC tasks. Higher (↑) is better for Audio-QA and AAC; lower (↓) is better for ASR. Datasets abbreviations are available in Table 7

Models	Size	Audio-QA (↑)				ASR (↓)						AAC (↑)	
		AE.	LQ.	TQA.	WQ.	Fzh.	A2.	A1.	WS_m.	WS_n.	Fen.	Clo.	MACS
Large Models (> 7B)													
Moshi	7B	25.63	48.30	16.75	16.85	-	-	-	-	-	-	4.32	12.01
SpeechGPT	7B	10.00	30.96	16.53	24.53	101.45	120.77	111.81	123.15	124.86	45.15	2.10	3.95
Kimi-Audio	7B	19.49	**57.53**	43.51	43.20	**2.87**	**2.53**	**0.61**	**6.34**	**5.39**	**4.87**	**55.92**	**64.90**
VITA-Audio	7B	40.20	54.30	18.59	30.75	6.35	5.56	4.58	20.38	15.88	9.58	6.18	7.94
LLaMA-Omni	8B	39.59	48.46	21.80	30.28	-	-	-	-	-	-	2.53	4.56
GLM-4-Voice	9B	**44.87**	**62.67**	**44.99**	**48.47**	158.47	425.84	414.77	207.14	270.21	223.07	13.15	12.67
Efficient Models (≤ 3B)													
Mini-Omni	0.5B	15.73	2.00	1.10	2.42	182.73	342.40	442.06	294.42	335.80	22.74	3.61	4.45
SLAM-Omni	0.5B	17.47	24.75	3.51	7.90	-	-	-	-	-	-	**54.52**	**50.46**
Qwen2.5-3B (AR)	3B	14.42	10.00	0.60	0.70	90.32	54.94	72.01	80.01	73.64	74.47	9.73	48.64
Qwen2.5-3B (NAR)	3B	11.31	0.67	1.21	0.70	68.94	212.27	160.58	89.22	111.29	83.51	9.54	27.40
TtT	3B	17.46	34.68	6.53	11.61	55.67	12.53	13.65	53.83	44.29	64.31	12.63	48.87
Pretrain+TtT	3B	**26.73**	**40.07**	**11.07**	**21.43**	**18.99**	**6.80**	**5.78**	**27.59**	**19.85**	**19.10**	11.55	42.86

Ablation Study To better understand the contribution of each training strategy in our hybrid AR-NAR framework, we perform an ablation study based on the full model TtT-3B (AR-NAR). The variant w/o BANOM corresponds to removing batchwise AR & NAR objective mixing from the full model, w/o PPM removes prefix preservation masking, and w/o SST removes stochastic span truncation. Table 1 ablation study part presents the detailed results of our ablation experiments. From these results we draw the following conclusions: (1) All three training strategies have a positive impact on model performance, and removing any one of them leads to clear degradation. For instance, on the LLaMAQuestions dataset, removing SST reduces the score from 34.68 to 10.20. This drop occurs because stochastic truncation mitigates positional bias in ⟨EOA⟩ prediction, forcing span termination by semantic content rather than position. Removing it weakens variable-length audio generation and reduces flexibility in conversational outputs. (2) Removing BANOM yields the largest performance degradation. For example, on the AISHELL-2 dataset, the performance decreases from 12.53 to 18.58 when the strategy is removed. This mechanism is essential for exposing text tokens to clean audio prefixes during training, better matching inference. Without it, the model faces sharper train–test discrepancy, weakening cross-modal consistency and alignment.

Effect of multimodal alignment pretraining. To further investigate the effectiveness of our method on top of a multimodally aligned pretrained model, we perform large-scale multimodal pretraining based on the Qwen2.5-3B-Base model. Specifically, we construct a corpus of approximately 200B tokens covering ASR, TTS, text-only data, and interleaved text–audio data. The model is trained with a standard AR objective using a global batch size of 256 for 140k steps. This pretraining stage equips the backbone model (Qwen2.5-3B-Base) with strong cross-modal alignment ability before applying our hybrid AR–NAR learning framework. Table 1 compares the AR-only and AR–NAR frameworks under two different training strategies, specifically training directly from Qwen2.5-3B-Base without multimodal pretraining (TtT-3B) and initialization from the multimodally aligned pretrained model (Pretrain+AR and Pretrain+TtT). From the table, we observe that: (1) When trained directly from Qwen2.5-3B-Base, our TtT framework achieves comparable or even superior performance to the AR-only baseline, indicating that the hybrid AR–NAR design is already competitive without pretraining; (2) when applied on top of the multimodally aligned pretrained model, Pretrain+TtT consistently matches or surpasses Pretrain+AR across both Audio-QA and ASR tasks. These results demonstrate that TtT not only performs strongly from scratch, but also benefits significantly when built upon large-scale multimodal alignment pretraining. Having validated the effectiveness of our hybrid architecture and the benefits of multimodal pretraining, we now compare our best model (Pretrain+TtT) against state-of-the-art audio-language models.

4.3 Benchmarking Against State-of-the-Art Models

Building on the demonstrated strengths of our hybrid AR–NAR architecture and multimodal pretraining, we now evaluate Pretrain+TtT against state-of-the-art audio-language models. Tables 2 and 3 group results by model scale, distinguishing efficient models from large ones. Notably,

Table 3: Evaluation results on URO-Bench. Higher (↑) is better for all tasks.

Models	**Size**	**Basic Task (↑)**			**Pro Task (↑)**			**Perceptual Quality (↑)**	
		Understanding	**Reasoning**	**Oral Conversation**	**Understanding**	**Reasoning**	**Oral Conversation**	**NMOS**	**UTMOS**
Large Models (> 7B)									
Moshi	7B	18.23	24.21	36.65	26.38	21.06	33.93	3.10	3.05
SpeechGPT	7B	9.26	13.34	35.50	19.03	14.29	28.88	4.04	3.92
Kimi-Audio	7B	83.89	53.88	54.44	53.25	41.44	50.17	3.52	2.93
VITA-Audio	7B	52.08	51.45	54.97	32.36	54.77	45.81	3.95	**4.24**
LLaMA-Omni	8B	53.71	41.93	64.05	34.66	51.51	43.91	**4.09**	4.00
GLM-4-Voice	9B	**85.82**	**61.63**	**69.90**	**55.47**	**51.89**	**61.30**	3.86	4.15
Efficient Models (≤ 3B)									
Mini-Omni	0.5B	15.01	14.80	29.71	23.51	33.09	33.46	4.15	4.42
SLAM-Omni	0.5B	31.55	26.45	42.20	34.49	27.39	40.23	**4.23**	**4.44**
Qwen2.5-3B (AR)	3B	34.32	13.15	23.68	16.32	34.99	25.90	3.96	4.16
Qwen2.5-3B (NAR)	3B	7.22	10.12	20.01	12.59	13.70	25.64	3.47	2.35
TtT	3B	43.39	24.00	30.08	23.37	33.78	34.82	3.89	4.25
Pretrain+TtT	3B	**57.63**	**39.30**	**45.68**	**32.38**	**43.76**	**46.10**	3.90	4.23

Moshi does not support ASR, and the official releases of LLaMA-Omni and SLAM-Omni lack ASR prompting, hence no ASR results are reported. GLM-4-Voice exhibits poor ASR performance due to the absence of task-specific system prompts. Mini-Omni and SpeechGPT exhibit poor generalization to Chinese ASR tasks, as they are trained solely on English speech. Among efficient models, Pretrain+TtT achieves state-of-the-art performance across Audio-QA, ASR, and AAC. It substantially outperforms 0.5B baselines such as Mini-Omni and SLAM-Omni on Audio-QA and ASR. While SLAM-Omni reports higher AAC scores (54.52 on Clotho, 50.46 on MACS), its official implementation relies on a separate 7B Vicuna model fine-tuned specifically for AAC. Notably, Pretrain+TtT also exceeds several 7B-scale models on some tasks: it outperforms SpeechGPT (7B) across all Audio-QA and ASR benchmarks, and surpasses Moshi (7B) on WQ. and AE. tasks. These results demonstrate that our hybrid AR–NAR design enables a compact 3B model to match or exceed the task-specific capabilities of significantly larger systems.

Beyond the standard benchmarks, we further validate our model on URO-Bench (Yan et al., 2025) a comprehensive speech-to-speech benchmark that assesses speech understanding, reasoning, and oral conversation across basic and pro difficulty levels. As shown in Table 3, among efficient models, Pretrain+TtT achieves the best performance across both basic and pro difficulty levels. Compared to large models, Pretrain+TtT outperforms Moshi (7B) and SpeechGPT (7B) across all task categories, and achieves comparable performance to VITA-Audio (7B) and LLaMA-Omni (8B) on pro reasoning tasks and pro oral conversation tasks. While GLM-4-Voice (9B) and Kimi-Audio (7B) achieve the highest scores overall, the performance gap is reasonable given their 3x model size. The perceptual quality of both TtT and Pretrain+TtT falls within the 3.89–4.5 range (NMOS & UTMOS), confirming consistently good audio synthesis quality. However, Kimi-Audio exhibits notably lower perceptual quality (UTMOS: 2.93, NMOS: 3.52) despite its strong task completion performance. This degradation stems from language consistency issues: Kimi-Audio frequently generates mixed Chinese-English audio or produces Chinese audio for English tasks. While the semantic content may be correct, such cross-lingual inconsistencies significantly degrade perceptual audio quality.

5 Conclusion

In this work, we introduce a unified framework that combines autoregressive text generation with non-autoregressive audio diffusion. By explicitly respecting the asymmetry between text and audio dependencies, our framework bridges the strengths of AR and NAR modeling within a single Transformer. We further propose simple yet effective strategies to mitigate train–test discrepancies, enabling robust and flexible audio generation. Experiments on Audio-QA and ASR benchmarks demonstrate clear improvements over strong AR and NAR baselines. Our results highlight the importance of modality-aware design for building scalable and effective speech-to-speech systems.

Reproducibility statement

The anonymous downloadable source code is available at: https://anonymous.4open.science/r/TtT. For theoretical results, a complete proof of the claim in included in the Section A.1.1 in Appendix.

References

Jacob Austin, Daniel D Johnson, Jonathan Ho, Daniel Tarlow, and Rianne Van Den Berg. Structured denoising diffusion models in discrete state-spaces. *Advances in Neural Information Processing Systems*, 34:17981–17993, 2021.

Zalán Borsos, Raphaël Marinier, Damien Vincent, Eugene Kharitonov, Olivier Pietquin, Matt Sharifi, Dominik Roblek, Olivier Teboul, David Grangier, Marco Tagliasacchi, et al. Audiolm: a language modeling approach to audio generation. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 31:2523–2533, 2023.

George EP Box, Gwilym M Jenkins, Gregory C Reinsel, and Greta M Ljung. *Time series analysis: forecasting and control*. John Wiley & Sons, 2015.

Junyi Chen, Shuming Shen, Andi Chen, Wen Wu, Jiantao Kang, Haohe Li, Weijiang Zhou, Yi Ren, Yanmin Qian, Xipeng Qiu, et al. Speecht5: Unified-modal encoder-decoder pre-training for spoken language processing. *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*, 2022.

Wenxi Chen, Ziyang Ma, Ruiqi Yan, Yuzhe Liang, Xiquan Li, Ruiyang Xu, Zhikang Niu, Yanqiao Zhu, Yifan Yang, Zhanxun Liu, Kai Yu, Yuxuan Hu, Jinyu Li, Yan Lu, Shujie Liu, and Xie Chen. Slam-omni: Timbre-controllable voice interaction system with single-stage training. In *Proceedings of Findings of the Association for Computational Linguistics*, Vienna, Austria, July 2025.

Yunfei Chu, Jin Xu, Qian Yang, Haojie Wei, Xipin Wei, Zhifang Guo, Yichong Leng, Yuanjun Lv, Jinzheng He, Junyang Lin, et al. Qwen2-audio technical report. *arXiv preprint arXiv:2407.10759*, 2024.

Trung Dang, David Aponte, Dung Tran, and Kazuhito Koishida. Livespeech: Low-latency zero-shot text-to-speech via autoregressive modeling of audio discrete codes. *arXiv preprint arXiv:2406.02897*, 2024.

Brian A Davey and Hilary A Priestley. *Introduction to lattices and order*. Cambridge university press, 2002.

Alexandre Défossez, Laurent Mazaré, Manu Orsini, Amélie Royer, Patrick Pérez, Hervé Jégou, Edouard Grave, and Neil Zeghidour. Moshi: a speech-text foundation model for real-time dialogue. *arXiv preprint arXiv:2410.00037*, 2024.

Ding Ding, Zeqian Ju, Yichong Leng, Songxiang Liu, Tong Liu, Zeyu Shang, Kai Shen, Wei Song, Xu Tan, Heyi Tang, et al. Kimi-audio technical report. *arXiv preprint arXiv:2504.18425*, 2025.

Konstantinos Drossos, Samuel Lipping, and Tuomas Virtanen. Clotho: An audio captioning dataset. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 736–740. IEEE, 2020.

Harishchandra Dubey, Ashkan Aazami, Vishak Gopal, Babak Naderi, Sebastian Braun, Ross Cutler, Alex Ju, Mehdi Zohourian, Min Tang, Mehrsa Golestaneh, et al. Icassp 2023 deep noise suppression challenge. *IEEE Open Journal of Signal Processing*, 5:725–737, 2024.

Qingkai Fang, Shoutao Guo, Yan Zhou, Zhengrui Ma, Shaolei Zhang, and Yang Feng. Llama-omni: Seamless speech interaction with large language models. In *Proceedings of the 13th International Conference on Learning Representations*, Singapore, April 2025.

Guhao Feng, Yihan Geng, Jian Guan, Wei Wu, Liwei Wang, and Di He. Theoretical benefit and limitation of diffusion language model. *arXiv preprint arXiv:2502.09622*, 2025.

Shansan Gong, Shivam Agarwal, Yizhe Zhang, Jiacheng Ye, Lin Zheng, Mukai Li, Chenxin An, Peilin Zhao, Wei Bi, Jiawei Han, et al. Scaling diffusion language models via adaptation from autoregressive models. *arXiv preprint arXiv:2410.17891*, 2024.

Ailin Huang, Boyong Wu, Bruce Wang, Chao Yan, Chen Hu, Chengli Feng, Fei Tian, Feiyu Shen, Jingbei Li, Mingrui Chen, et al. Step-audio: Unified understanding and generation in intelligent speech interaction. *arXiv preprint arXiv:2502.11946*, 2025.

Jungil Kong, Jaehyeon Kim, and Jaekyoung Bae. Hifi-gan: Generative adversarial networks for efficient and high fidelity speech synthesis. *Advances in Neural Information Processing Systems*, 33:17022–17033, 2020.

Gen Li and Changxiao Cai. A convergence theory for diffusion language models: An information-theoretic perspective. *arXiv preprint arXiv:2505.21400*, 2025.

Tianpeng Li, Jun Liu, Tao Zhang, Yuanbo Fang, Da Pan, Mingrui Wang, Zheng Liang, Zehuan Li, Mingan Lin, Guosheng Dong, et al. Baichuan-audio: A unified framework for end-to-end speech interaction. *arXiv preprint arXiv:2502.17239*, 2025.

Alexander H Liu, Sang-gil Lee, Chao-Han Huck Yang, Yuan Gong, Yu-Chiang Frank Wang, James R Glass, Rafael Valle, and Bryan Catanzaro. Uniwav: Towards unified pre-training for speech representation learning and generation. *arXiv preprint arXiv:2503.00733*, 2025.

Zuwei Long, Yunhang Shen, Chaoyou Fu, Heting Gao, Lijiang Li, Peixian Chen, Mengdan Zhang, Hang Shao, Jian Li, Jinlong Peng, et al. Vita-audio: Fast interleaved cross-modal token generation for efficient large speech-language model. *arXiv preprint arXiv:2505.03739*, 2025.

Justin Lovelace, Varsha Kishore, Yiwei Chen, and Kilian Weinberger. Diffusion guided language modeling. *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 14936–14952, 2024.

Irene Martín-Morató and Annamaria Mesaros. What is the ground truth? reliability of multi-annotator data for audio tagging. In *2021 29th European Signal Processing Conference (EUSIPCO)*, pp. 76–80. IEEE, 2021.

Shivam Mehta, Ruibo Tu, Jonas Beskow, Éva Székely, and Gustav Eje Henter. Matcha-tts: A fast tts architecture with conditional flow matching. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 11341–11345. IEEE, 2024.

Shen Nie, Fengqi Zhu, Zebin You, Xiaolu Zhang, Jingyang Ou, Jun Hu, Jun Zhou, Yankai Lin, Ji-Rong Wen, and Chongxuan Li. Large language diffusion models. *arXiv preprint arXiv:2502.09992*, 2025.

Jingyang Ou, Shen Nie, Kaiwen Xue, Fengqi Zhu, Jiacheng Sun, Zhenguo Li, and Chongxuan Li. Your absorbing discrete diffusion secretly models the conditional distributions of clean data. *arXiv preprint arXiv:2406.03736*, 2024.

Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. Robust speech recognition via large-scale weak supervision, 2022. URL `https://arxiv.org/abs/2212.04356`.

Marc'Aurelio Ranzato, Sumit Chopra, Michael Auli, and Wojciech Zaremba. Sequence level training with recurrent neural networks. *arXiv preprint arXiv:1511.06732*, 2015.

Yi Ren, Jinglin Liu, Xu Tan, Zhou Zhao, Sheng Zhao, and Tie-Yan Liu. A study of non-autoregressive model for sequence generation. *arXiv preprint arXiv:2004.10454*, 2020.

Paul K Rubenstein, Chulayuth Asawaroengchai, Duc Dung Nguyen, Ankur Bapna, Zalán Borsos, Félix de Chaumont Quitry, Peter Chen, Dalia El Badawy, Wei Han, Eugene Kharitonov, et al. Audiopalm: A large language model that can speak and listen. *arXiv preprint arXiv:2306.12925*, 2023.

Takaaki Saeki, Detai Xin, Wataru Nakata, Tomoki Koriyama, Shinnosuke Takamichi, and Hiroshi Saruwatari. Utmos: Utokyo-sarulab system for voicemos challenge 2022. *arXiv preprint arXiv:2204.02152*, 2022.

Subham Sahoo, Marianne Arriola, Yair Schiff, Aaron Gokaslan, Edgar Marroquin, Justin Chiu, Alexander Rush, and Volodymyr Kuleshov. Simple and effective masked diffusion language models. *Advances in Neural Information Processing Systems*, 37:130136–130184, 2024.

Jiaxin Shi, Kehang Han, Zhe Wang, Arnaud Doucet, and Michalis Titsias. Simplified and generalized masked diffusion for discrete data. *Advances in Neural Information Processing Systems*, 37: 103131–103167, 2024.

Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, Katie Millican, et al. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023.

Chengyi Wang, Sanyuan Chen, Yu Wu, Ziyang Zhang, Long Zhou, Shujie Liu, Zhuo Chen, Yanqing Liu, Huaming Wang, Jinyu Li, et al. Neural codec language models are zero-shot text to speech synthesizers. *arXiv preprint arXiv:2301.02111*, 2023.

Tsung-Han Wu, Joseph E Gonzalez, Trevor Darrell, and David M Chan. Clair-a: Leveraging large language models to judge audio captions. *arXiv preprint arXiv:2409.12962*, 2024.

Zhifei Xie and Changqiao Wu. Mini-omni: Language models can hear, talk while thinking in streaming. *arXiv preprint arXiv:2408.16725*, 2024a.

Zhifei Xie and Changqiao Wu. Mini-omni: Language models can hear, talk while thinking in streaming. *arXiv preprint arXiv:2408.16725*, 2024b.

Jin Xu, Zhifang Guo, Jinzheng He, Hangrui Hu, Ting He, Shuai Bai, Keqin Chen, Jialin Wang, Yang Fan, Kai Dang, et al. Qwen2. 5-omni technical report. *arXiv preprint arXiv:2503.20215*, 2025.

Ruiqi Yan, Xiquan Li, Wenxi Chen, Zhikang Niu, Chen Yang, Ziyang Ma, Kai Yu, and Xie Chen. Uro-bench: A comprehensive benchmark for end-to-end spoken dialogue models. *arXiv preprint arXiv:2502.17810*, 2025.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025a.

Ling Yang, Ye Tian, Bowen Li, Xinchen Zhang, Ke Shen, Yunhai Tong, and Mengdi Wang. Mmada: Multimodal large diffusion language models. *arXiv preprint arXiv:2505.15809*, 2025b.

Jiacheng Ye, Jiahui Gao, Shansan Gong, Lin Zheng, Xin Jiang, Zhenguo Li, and Lingpeng Kong. Beyond autoregression: Discrete diffusion for complex reasoning and planning. *arXiv preprint arXiv:2410.14157*, 2024.

Runpeng Yu, Xinyin Ma, and Xinchao Wang. Dimple: Discrete diffusion multimodal large language model with parallel decoding. *arXiv preprint arXiv:2505.16990*, 2025.

Aohan Zeng, Zhengxiao Du, Mingdao Liu, Kedong Wang, Shengmin Jiang, Lei Zhao, Yuxiao Dong, and Jie Tang. Glm-4-voice: Towards intelligent and human-like end-to-end spoken chatbot. *arXiv preprint arXiv:2412.02612*, 2024.

Aohan Zeng, Zhengxiao Du, Mingdao Liu, Lei Zhang, Shengmin Jiang, Yuxiao Dong, and Jie Tang. Scaling speech-text pre-training with synthetic interleaved data. In *Proceedings of the 13th International Conference on Learning Representations*, Singapore, April 2025.

Dong Zhang, Shimin Li, Xin Zhang, Jun Zhan, Pengyu Wang, Yaqian Zhou, and Xipeng Qiu. Speechgpt: Empowering large language models with intrinsic cross-modal conversational abilities. *arXiv preprint arXiv:2305.11000*, 2023.

A Appendix

A.1 Mathematical Derivation

A.1.1 Derivation of the Training Objective Upper Bound

Recall from Eq. 5 that the joint distribution factors as:

$$\tilde{p}_\theta(x) = \prod_{m=1}^{M} \left[\prod_{j=1}^{|\mathcal{T}_m|} p_\theta\big(t_{m,j} \mid \mathcal{T}_{<m}, \mathcal{A}_{<m}, t_{m,<j}\big) \cdot \tilde{p}_\theta(\mathcal{A}_m \mid \mathcal{T}_{\leq m}, \mathcal{A}_{<m}) \right]. \tag{9}$$

Taking the negative logarithm of both sides gives:

$$\begin{aligned} -\log \tilde{p}_\theta(x) &= -\sum_{m=1}^{M} \sum_{j=1}^{|\mathcal{T}_m|} \log p_\theta\big(t_{m,j} \mid \mathcal{T}_{<m}, \mathcal{A}_{<m}, t_{m,<j}\big) - \sum_{m=1}^{M} \log \tilde{p}_\theta(\mathcal{A}_m \mid \mathcal{T}_{\leq m}, \mathcal{A}_{<m}) \\ &= \mathcal{L}_{\text{AR}}(x) + \sum_{m=1}^{M} \left(-\log \tilde{p}_\theta(\mathcal{A}_m \mid \mathcal{T}_{\leq m}, \mathcal{A}_{<m})\right). \end{aligned} \tag{10}$$

By Eq. 6 and its summation over m, we have:

$$\mathcal{L}_{\text{AO}}(x) \geq \sum_{m=1}^{M} \left(-\log \tilde{p}_\theta(\mathcal{A}_m \mid \mathcal{T}_{\leq m}, \mathcal{A}_{<m})\right). \tag{11}$$

Therefore, combining both components:

$$\mathcal{L}_{\text{AR}}(x) + \mathcal{L}_{\text{AO}}(x) \geq \mathcal{L}_{\text{AR}}(x) + \sum_{m=1}^{M} \left(-\log \tilde{p}_\theta(\mathcal{A}_m \mid \mathcal{T}_{\leq m}, \mathcal{A}_{<m})\right) = -\log \tilde{p}_\theta(x), \tag{12}$$

which establishes Eq. 8. This confirms that our practical training objective $\mathcal{L}_{\text{Unified}}(x)$ is a valid upper bound on the true negative log-likelihood, enabling tractable optimization while preserving consistency with the target joint distribution $\tilde{p}_\theta(x)$.

A.2 Related Work

A.2.1 Audio-Language Model Pretraining

Recent advances in end-to-end audio-language models have moved beyond traditional cascaded architectures (Chen et al., 2022; Wang et al., 2023) toward unified multimodal frameworks. Representative works include Moshi (Défossez et al., 2024), which achieves real-time duplex speech conversation through hierarchical Transformer architectures; GLM4-Voice (Zeng et al., 2024), which builds upon GLM-4-9B for robust Chinese and English speech processing; and VITA-Audio (Long et al., 2025), which introduces a lightweight Multiple Cross-modal Token Prediction (MCTP) module for fast audio-text generation with significantly reduced first-token latency. More recent efforts have focused on scaling and production readiness: Step-Audio (Huang et al., 2025) presents a 130B-parameter unified speech-text model with generative speech data engine and instruction-driven fine control across dialects, emotions, singing, and RAP, while Baichuan-Audio (Li et al., 2025) features text-guided aligned speech generation with multi-codebook discretization to preserve both semantic and acoustic information. UniWav (Liu et al., 2025) proposes the first unified encoder-decoder framework that jointly learns representation encoders and generative audio decoders for both discriminative and generative speech tasks.

A key limitation shared by these approaches is their reliance on uniform autoregressive objectives for both text and audio tokens, which overlooks the distinct dependency structures of these modalities. Our work addresses this gap by proposing a hybrid AR-NAR framework that respects the inherent asymmetries between text and audio generation.

A.2.2 Discrete Diffusion Models

Discrete diffusion models have emerged as a compelling alternative to autoregressive generation, offering non-autoregressive approaches that can generate entire sequences in parallel. The foundational work of D3PMs (Austin et al., 2021) generalized diffusion processes to discrete data through flexible transition matrices, with absorbing processes that progressively mask tokens proving particularly effective. This framework has since evolved through both theoretical advances and practical improvements. From a theoretical perspective, recent work has deepened our understanding of discrete diffusion dynamics. Ou et al. (2024) revealed that absorbing diffusion's concrete score can be expressed as time-independent conditional probabilities, leading to RADD—a reparameterized model that removes explicit time conditioning while establishing connections to any-order autoregressive generation. Building on this foundation, Li & Cai (2025) formally characterized convergence rates, proving that KL divergence decays at $O(1/T)$ with bounds scaling linearly with token mutual information. However, Feng et al. (2025) identified a fundamental trade-off: while masked diffusion achieves near-optimal perplexity in constant steps, sequence-level tasks like reasoning may require steps linear in sequence length. Practical advances have focused on training efficiency and application domains. Shi et al. (2024) reformulated the variational objective as a weighted integral of cross-entropy losses, unifying prior approaches while achieving state-of-the-art results that even surpass comparable autoregressive baselines. For complex reasoning tasks where autoregressive models struggle with subgoal imbalance, Ye et al. (2024) demonstrated that Multi-Granularity Diffusion Modeling can achieve near-perfect accuracy by prioritizing harder subgoals during training. The scalability challenge has been addressed through innovative adaptation strategies. Rather than training from scratch, Gong et al. (2024); Nie et al. (2025) showed that pretrained autoregressive models can be efficiently converted to diffusion models via continual pre-training, maintaining competitive performance while enabling parallel generation. Meanwhile, hybrid approaches are gaining traction: Lovelace et al. (2024) combined diffusion-based latent proposals with autoregressive decoding for controllable generation, while Yang et al. (2025b) developed MMaDA, a unified multimodal diffusion foundation model that processes text, images, and reasoning within a single architecture.

A.3 Autoregressive Modeling & Absorbing Discrete Diffusion

A.3.1 Autoregressive Modeling

Autoregressive (AR) models are a fundamental class of generative models that factorize the joint probability distribution of a sequence $x = (x^1, \dots, x^L)$ into a product of conditional probabilities, based on the chain rule:

$$p(x) = \prod_{i=1}^{L} p(x^i | x^{<i}) \tag{13}$$

where $x^{<i} = (x^1, \dots, x^{i-1})$ represents the tokens preceding the current token x^i. This factorization imposes a sequential, causal structure on the generation process. Such models, typically implemented with Transformer decoders, are trained by minimizing the negative log-likelihood (NLL) of the data, which corresponds to a cross-entropy loss at each position.

A.3.2 Absorbing Discrete Diffusion

Discrete diffusion models offer a non-autoregressive alternative for sequence generation. We focus on absorbing discrete diffusion (Austin et al., 2021; Ou et al., 2024), which involves a forward corruption process and a learned reverse denoising process.

Forward Process. The forward process is a continuous-time discrete Markov chain that corrupts a clean sequence x_0 over a time interval $t \in [0, T]$. Its dynamics are governed by a time-dependent transition rate matrix $\boldsymbol{Q}_t = \sigma(t)\boldsymbol{Q}$, where $\sigma(t)$ is a positive noise schedule. For absorbing diffusion, the constant matrix $\boldsymbol{Q} = \boldsymbol{Q}^{\text{abs}}$ is defined as:

$$\boldsymbol{Q}^{\text{abs}}(x \to x') = \begin{cases} 1, & \text{if } x' = [\mathbf{M}] \text{ and } x \neq [\mathbf{M}], \\ -1, & \text{if } x' = x \neq [\mathbf{M}], \\ 0, & \text{otherwise.} \end{cases} \tag{14}$$

This structure dictates that any token $x \neq [\mathbf{M}]$ transitions to a special mask token $[\mathbf{M}]$ at a rate of $\sigma(t)$. The state $[\mathbf{M}]$ is an **absorbing state** because the transition rate out of it is zero (i.e., $\boldsymbol{Q}^{\text{abs}}([\mathbf{M}] \to x') = 0$ for all x'), meaning once a token is masked, it remains masked. Over time, the sequence converges to a fully masked state. The probability that a token is masked by time t is given by $\lambda(t) = 1 - e^{-\int_0^t \sigma(s)ds}$.

Reverse Process. The reverse process is also a continuous-time Markov chain that learns to denoise a corrupted sequence x_t back towards the clean data x_0. Its reverse transition rate matrix $\tilde{\boldsymbol{Q}}_t$ is related to the forward rate matrix by:

$$\tilde{\mathbf{Q}}_t(x_t \to \hat{x}_t) = \begin{cases} \mathbf{Q}_t(\hat{x}_t \to x_t)\frac{p_t(\hat{x}_t)}{p_t(x_t)}, & x_t \neq \hat{x}_t, \\ -\sum_{k \neq x} \tilde{\mathbf{Q}}_t(x_t, k), & \hat{x}_t = x_t. \end{cases} \tag{15}$$

The term $p_t(\hat{x}_t)/p_t(x_t)$ is known as the *concrete score*. Since the forward process only allows transitions to the $[\mathbf{M}]$ state, the only non-trivial reverse transitions are from $[\mathbf{M}]$ back to a vocabulary token. This simplifies the learning task to modeling the score for these specific denoising transitions.

Time-Independent Score and the Denoising Objective. A key theoretical insight for absorbing diffusion is that the concrete score analytically decomposes into a known, time-dependent scalar and a *time-independent* conditional probability over the clean data (Ou et al., 2024). Specifically, for a transition that unmasks position i from $[\mathbf{M}]$ to a token v, the score is:

$$\underbrace{\frac{p_t(\ldots, \hat{x}^i = v, \ldots)}{p_t(\ldots, x^i = [\mathbf{M}], \ldots)}}_{\text{concrete score}} = \underbrace{\frac{e^{-\bar{\sigma}(t)}}{1 - e^{-\bar{\sigma}(t)}}}_{\text{time scalar}} \cdot \underbrace{p_0(v \mid UM)}_{\text{clean conditional probability}} . \tag{16}$$

where UM denotes the set of unmasked (visible) tokens in the corrupted sequence. This decomposition is crucial because it decouples the time-dependent dynamics from the data distribution. It implies that the model q_θ does not need to learn a complex function of time t. Instead, its sole objective is to learn to approximate the clean conditional distribution $p_0(v|UM)$, which is a static, time-independent property of the data. The learning task is thus simplified to a denoising objective: given a corrupted sequence with some tokens masked, predict the original tokens for the masked positions based on the visible context.

Equivalence to Any-Order Autoregressive Modeling. This denoising perspective reveals a profound connection to autoregressive modeling. A standard AR model learns to predict a token based on a fixed, causal context. The diffusion model, through its denoising objective, learns to predict a token given an arbitrary context of unmasked tokens. This ability to condition on any subset of the context is the defining feature of an Any-Order Autoregressive Model (AO-ARM).

In fact, the principled training objective for the diffusion model, known as the λ-denoising cross-entropy loss, is mathematically equivalent to the training objective of an AO-ARM (Ou et al., 2024), which averages the prediction loss over all possible permutations (or orderings) of the sequence:

$$\mathcal{L}_{AO}(\boldsymbol{x}_0) = \mathbb{E}_{\pi \sim U_\pi} \sum_{l=1}^{d} -\log q_\theta(x_0^{\pi(l)} | x_0^{\pi(<l)}). \tag{17}$$

where π is a random permutation of the token indices. Therefore, training an absorbing discrete diffusion model is equivalent to training a powerful ensemble of autoregressive models that can operate in any order. This inherent flexibility is what enables parallel, non-autoregressive generation at inference time and makes it a suitable choice for modeling source-dependent modalities like audio.

A.4 Block-wise Masked Diffusion Generation For Audio Tokens

For NAR audio generation, we employ a block-wise denoising approach adapted from Nie et al. (2025). Unlike full-sequence diffusion, it processes audio in fixed-length blocks, balancing parallelism and controllability.

As detailed in Algorithm 1, the model generates audio in fixed-size blocks of length B, where each block is progressively denoised over T steps using an absorbing discrete diffusion process.

At each denoising step t, the model predicts tokens for all currently masked positions in parallel. The algorithm then selectively commits the most confident predictions (determined by predicted probability or random sampling) while remasking the remaining positions for further refinement. This progressive denoising continues until all positions in the current block are decoded. Crucially, if an ⟨EOA⟩ token is generated within a block, decoding terminates immediately at that position, truncating the remainder and seamlessly returning control to the AR text generation mode.

Algorithm 1 Block-wise Masked Diffusion for Autoregressive Audio Generation

```
Require: Context tokens $\mathbf{c} \in \mathbb{N}^{1\times L_c}$, max generation length $L_{\max} \in \mathbb{N}$,
  Sampling steps $T \in \mathbb{N}$, block length $B \in \mathbb{N}$, temperature $\tau \geq 0$,
  CFG scale $\gamma \geq 0$, remasking strategy $\mathcal{R} \in \{\texttt{low\_confidence}, \texttt{random}\}$,
  Special token IDs: mask $m_{\text{mask}}$, end-of-audio $\mathcal{E}$.
Ensure: Generated token sequence $\mathbf{s} \in \mathbb{N}^{1\times L}$ with $L \leq L_c + L_{\max}$.
  Initialize $\mathbf{s} \leftarrow \mathbf{c}$                                ▷ Start from context
  while $|\mathbf{s}| < |\mathbf{c}| + L_{\max}$ do
      $\mathbf{x} \leftarrow \texttt{pad}(\mathbf{s}, B, \text{value} = m_{\text{mask}})$          ▷ Append $B$ mask tokens
      $\mathcal{M}_{\text{block}} \leftarrow \{i \mid \mathbf{x}_i = m_{\text{mask}} \land i \geq |\mathbf{s}|\}$       ▷ Masked block indices
      $\{n_t\}_{t=1}^{T} \leftarrow \texttt{schedule}(|\mathcal{M}_{\text{block}}|, T)$          ▷ Tokens to decode per step
      for $t = 1$ to $T$ do
          $\mathcal{M}_t \leftarrow \{i \mid \mathbf{x}_i = m_{\text{mask}}\}$                 ▷ Current mask positions
          if $\gamma > 0$ then
              $\mathbf{x}_{\text{uncond}} \leftarrow \mathbf{x}$; $\mathbf{x}_{\text{uncond}}[\neg\mathcal{M}_t] \leftarrow m_{\text{mask}}$          ▷ Unconditional input
              $\boldsymbol{\ell}_{\text{cond}}, \boldsymbol{\ell}_{\text{uncond}} \leftarrow \texttt{model}([\mathbf{x}; \mathbf{x}_{\text{uncond}}])$          ▷ Batched forward
              $\boldsymbol{\ell} \leftarrow \boldsymbol{\ell}_{\text{uncond}} + (\gamma + 1) \cdot (\boldsymbol{\ell}_{\text{cond}} - \boldsymbol{\ell}_{\text{uncond}})$
          else
              $\boldsymbol{\ell} \leftarrow \texttt{model}(\mathbf{x})$
          end if
          $\hat{\mathbf{x}} \leftarrow \arg\max(\texttt{Gumbel}(\boldsymbol{\ell}, \tau))$                ▷ Gumbel sampling
          if $\mathcal{R} = \texttt{low\_confidence}$ then
              $\mathbf{p} \leftarrow \texttt{softmax}(\boldsymbol{\ell})$; $\mathbf{c}_i \leftarrow \mathbf{p}_i[\hat{\mathbf{x}}_i]$          ▷ Confidence = predicted prob
          else if $\mathcal{R} = \texttt{random}$ then
              $\mathbf{c}_i \leftarrow \texttt{Uniform}(0, 1)$ for $i \in \mathcal{M}_t$
          end if
          $\mathbf{c}_i \leftarrow -\infty$ for $i < |\mathbf{s}|$                     ▷ Protect context tokens
          $\hat{\mathbf{x}}_i \leftarrow \mathbf{x}_i$ for $i \notin \mathcal{M}_t$                  ▷ Only update masked positions
          $\mathcal{K}_t \leftarrow \texttt{TopK}(\{\mathbf{c}_i \mid i \in \mathcal{M}_t\}, k = n_t)$     ▷ Select $n_t$ most confident/random tokens
          $\mathbf{x}_i \leftarrow \hat{\mathbf{x}}_i$ for all $i \in \mathcal{K}_t$                   ▷ Commit tokens to sequence
      end for
      $\mathbf{b} \leftarrow \mathbf{x}[|\mathbf{s}| : |\mathbf{s}| + B]$                       ▷ Extract generated block
      if $\mathcal{E} \cap \mathbf{b} \neq \emptyset$ then
          $p \leftarrow \min\{i \mid \mathbf{b}_i \in \mathcal{E}\}$; $\mathbf{s} \leftarrow [\mathbf{s}, \mathbf{b}_{:p+1}]$; return $\mathbf{s}$   ▷ Early termination at first end token
      end if
      $\mathbf{s} \leftarrow [\mathbf{s}, \mathbf{b}]$                              ▷ Append full block
  end while
  return $\mathbf{s}$
```

A.5 Illustration of Training Loss and Attention Design

Figure 3(a) shows the training loss of our framework: AR loss is applied to text spans, while NAR loss—based on discrete diffusion—is used for audio spans. Although we employ discrete diffusion, our framework is extensible to other NAR generation methods. Figure 3(b) visualizes the attention pattern described in Section 3.4.

A.6 Dataset details

Table 4 provides a summary of the training datasets, with detailed examples provided in the Appendix A.9.1. During training, we aim to construct a balanced corpus that supports effective learning

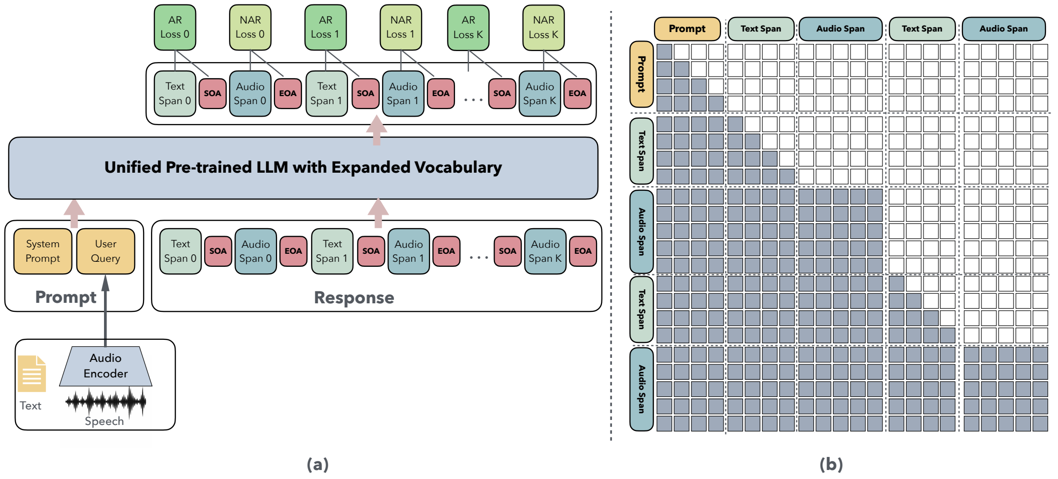


Figure 3: **Training loss and attention design. (a) Training pipeline.** Starting from a pretrained text LLM, we expand the vocabulary with audio tokens and control symbols. Text spans use AR cross-entropy loss while audio spans use NAR diffusion loss, sharing a single Transformer backbone. **(b) Attention pattern.** Text spans follow causal attention (left-to-right), while audio spans use bidirectional attention within spans but causal attention across spans, enabling parallel audio generation while preserving cross-modal dependencies.

across multiple tasks. Specifically, we randomly sample one million instances from the ASR dataset, the TTS dataset, and the audio chat dataset respectively. In addition, we create bilingual interleaved text and audio data, ensuring that Chinese and English are represented in approximately equal proportions. To build the audio chat corpus, we rely on the text-to-audio dataset VoiceAssistant-400K together with the text-based datasets OpenHermes-2.5 and Firefly-Train-1.1M, and we employ a TTS model, namely CosyVoice2, to convert text into synthetic audio so as to enrich the training data. To further enhance cross-modal alignment between text and audio, we follow prior work (Zeng et al., 2025) and supplement the training corpus with interleaved text and audio data derived from the large-scale pretrained corpus FineWeb-Edu. This strategy not only expands task coverage but also strengthens the model's ability to jointly learn from and align textual and acoustic modalities. The evaluation datasets are shown in Table 7.

A.7 Evaluation Deatils

A.7.1 Evaluation Tasks

URO-Bench We leverage URO-Bench Yan et al. (2025) for a more comprehensive evaluation of our proposed method against existing baselines. URO-Bench is specifically designed for audio-in, audio-out tasks, directly simulating real-world conversational scenarios. In this framework, the spoken outputs from each model are first transcribed into text using Whisper-large Radford et al. (2022), and the resulting transcripts are then evaluated for correctness, coherence, and task alignment. This evaluation employs a hybrid scoring framework comprising three components: (1) an LLM-as-a-judge (originally implemented via commercial LLM APIs) to assess semantic correctness and task alignment; (2) rule-based metrics for automatic Word Error Rate (WER) computation; and (3) a fine-tuned emotion-aware model to evaluate the appropriateness of affective expression in spoken responses. Together, these components ensure that model outputs are judged not merely on surface-level textual fidelity, but on semantic accuracy, transcription quality, and emotional coherence.

URO-Bench structures its evaluation across two difficulty levels: Basic and Pro. Each level comprises three distinct categories: Understanding tasks, Reasoning tasks, and Oral Conversation tasks. This hierarchical design facilitates a fine-grained assessment of model capabilities across increasing levels of linguistic and cognitive demand, covering both single-round and multi-round scenarios. In our experiments, we report results on the English subset of URO-Bench's evaluation set. Due to limited access to the original evaluation APIs (Gemini Flash and GPT-4o-mini), we substitute the

Table 4: Summary of datasets used in training.

Dataset	Language	Samples	Task Type
Emilia_zh	Chinese	500000	TTS
Emilia_en	English	500000	TTS
AISHELL2	Chinese	600000	ASR
AISHELL3	Chinese		ASR
CommonVoice	Chinese, English		ASR
GigaSpeech	English		ASR
LibriSpeech	English		ASR
MLS-Eng	English		ASR
PeopleSpeech	English		ASR
VoxPopuli	English		ASR
WenetSpeech	Chinese	400000	ASR
VoiceAssistant-400K	English	1000000	Audio Chat
OpenHermes-2.5	English		Audio Chat
Firefly-Train-1.1M	Chinese		Audio Chat
MathInstruct	English	262039	Text Chat
MACS	English	59282	AAC
Clotho-v2	English		AAC
Nonspeech7k	English		SEC
VocalSound	English		SEC
CochlScene	English		ASC
Chinese-Fineweb-Edu (Skypile)	Chinese	1500000	Interleaved Data
FineWeb-Edu	English	1500000	Interleaved Data
Total	–	**6321321**	

LLM-as-a-judge component with thinking model Qwen3-30B-A3B, while keeping all other components—including the Whisper-based ASR pipeline and rule-based scoring—identical to the original implementation.

Furthermore, the benchmark integrates a perceptual quality evaluation mechanism. We employ the strong UTMOS Saeki et al. (2022) for the UTMOS score and DNSMOS Dubey et al. (2024) for the NMOS score evaluation, enabling the joint assessment of content accuracy and acoustic quality. Importantly, none of the URO-Bench data was used during training or validation, ensuring an unbiased assessment of generalization.

Audio-QA Task In addition to URO-Bench, we also evaluate our model with the Audio-QA tasks established by Kimi-Audio Ding et al. (2025). Previous evaluation framework in Kimi-Audio assess Audio-QA performance using the text portion of interleaved outputs, which overlooks the fact that the audio output of an end-to-end speech model more directly reflects its ability to generate natural and semantically faithful responses. To address this limitation, we evaluate Audio-QA directly on the audio outputs of our framework by first applying an ASR model to transcribe the generated audio into text, where Whisper-Large-v3 Radford et al. (2022) is used for English audio and Paraformer-zh for Chinese audio, with a comparison of ASR performance across different models provided in Table 5. The transcribed text is then combined with the original QA queries and the ground truth answers and passed to a large scale reasoning model, Qwen3-235B-A30B, which serves as an LLM-as-a-Judge model to determine whether the response semantically matches the reference and to provide either a correctness label or a graded score. We report the average accuracy or score on the benchmark, and this evaluation pipeline provides a more faithful assessment of our model's audio-to-audio QA ability in realistic conversational scenarios where speech serves as the output modality.

ASR Task To assess the model's capability in aligning speech with textual representations, we evaluate it on the ASR task, where the model generates text transcriptions from input audio and performance is measured using word error rate (WER). A lower WER indicates more accurate recogni-

Table 5: WER performance of different ASR models on Chinese (zh) and English (en).

Model	WER-zh (↓)	WER-en (↓)
Whisper-Large-v3	0.5054	**0.2167**
Paraformer-zh	**0.1028**	0.3946

tion, which reflects not only strong ASR ability but also effective cross modal consistency achieved by our hybrid AR-NAR modeling framework.

AAC Task To assess the model's capacity to comprehend complex or acoustically challenging audios, we evaluate its audio captioning (AAC) performance on two established benchmarks: Clotho-v2 Drossos et al. (2020) and MACS Martín-Morató & Mesaros (2021). We input audio clips and generate corresponding textual captions. The quality of these captions is then evaluated using Qwen3-30B-A3B Yang et al. (2025a), guided by the evaluation prompt introduced in CLAIR-A Wu et al. (2024), which emphasizes semantic relevance, completeness, and naturalness. The judge assigns a score on a 0–100 scale, where higher scores indicate better caption quality.

A.7.2 Evaluation Datasets

URO-Bench We use the English portion of URO-Bench Yan et al. (2025) to evaluate our model's performance. As detailed in Table 6, the benchmark consists of 10 basic tasks and 12 pro tasks. The basic tasks include 4 oral conversation, 4 reasoning, and 2 understanding tasks, while the pro tasks comprise 4 understanding, 4 reasoning, and 4 oral conversation tasks. The final score is obtained by first averaging the model's performance on each dataset, and then averaging these scores within each (difficulty, category) group.

Table 6: Evaluation datasets used from URO-Bench.

Dataset	Task /Evaluation Aspect	data nums	Category
Basic tasks			
AlpacaEval	Authentic, open-ended dialogue	199	Oral Conversation
CommonEval	Authentic, open-ended dialogue	200	Oral Conversation
WildchatEval	Real-world conversation	349	Oral Conversation
StoralEval	Deduce morals from a given story	201	Reasoning
Summary	Summarize a given story or statement	118	Oral Understanding
TruthfulEval	Factual questions about life	470	Reasoning
GaokaoEval	English listening questions	303	Understanding
Gsm8kEval	Practical mathematical problems	582	Reasoning
MLC	Mathematics, logic, and common sense	177	Reasoning
Repeat	Repeat the user's words verbatim	252	Understanding
Pro Tasks			
CodeSwitching-en	Understand code switching sentences	70	Understanding
GenEmotion-en	Respond in a specified tone	54	Oral Conversation
GenStyle-en	Respond in a specified style	44	Oral Conversation
MLCpro	Difficult mathematical, scientific questions	91	Reasoning
Safety-en	Reject answering privacy-related questions	24	Reasoning
SRT-en	Sing, recite poems, read tongue twisters	43	Oral Conversation
UnderEmotion-en	Understand the speaker's mood	137	Understanding
Multilingual	Respond in multiple languages	1108	Oral Conversation
ClothoEval-en	Comprehension of general ambient sounds	265	Understanding
MuChoEval-en	Comprehension of music	311	Understanding
MtBenchEval-en	Multi-round spoken dialogue	190	Reasoning
SpeakerAware-en	Multi-speaker multi-round dialogues	55	Reasoning

Audio-QA, ASR and AAC Task We evaluate model performance on a diverse set of benchmarks covering both Audio Question Answering (Audio-QA), Automatic Speech Recognition (ASR), and automatic audio caption (AAC) tasks. For Audio-QA, we use four datasets: AlpacaEval, TriviaQA, and WebQuestions (English), along with LLaMAQuestions (English), assessing cross-lingual reasoning and comprehension from speech. For ASR, we include five datasets: Fleurs-zh/en (multilingual), AISHELL-1/2, and WenetSpeech (all Chinese), covering varied domains, accents, and recording conditions to robustly measure transcription accuracy. For AAC, we include two datasets: Clotho-v2 (English) and MACS (English), covering natural audios collected from various environmental sound clips, not limited to human-to-human dialogue, which helps assess the model's ability to understand the environment rather than simple language processing. Dataset details are summarized in Table 7.

Table 7: Evaluation datasets used for Audio-QA, ASR and AAC tasks.

Dataset	**Language**	**Task Type**	**Abbreviation**
AlpacaEval	English	Audio-QA	AE.
LLaMAQuestions	English	Audio-QA	LQ.
TriviaQA	English	Audio-QA	TQA.
WebQuestions	English	Audio-QA	WQ.
Fleurs-zh	Chinese	ASR	Fzh.
AISHELL-2	Chinese	ASR	A2.
AISHELL-1	Chinese	ASR	A1.
WenetSpeech-test_meeting	Chinese	ASR	WS_m.
WenetSpeech-test_net	Chinese	ASR	WS_n.
Fleurs-en	English	ASR	Fen.
Clotho-v2	English	AAC	Clo.
MACS	English	AAC	MACS

A.8 Baselines

We compare our TtT model with the following state-of-the-art large audio-language models to evaluate its effectiveness:

- Moshi Défossez et al. (2024): It unifies streaming speech and text understanding within a single autoregressive framework, aligning acoustic and linguistic representations for low-latency real-time dialogue and robust multimodal instruction following.
- SpeechGPT Zhang et al. (2023): It incorporates discrete speech tokens into a single language model and follows a three-stage training pipeline to enable unified speech–text understanding and cross-modal instruction following within one framework.
- Kimi-Audio Ding et al. (2025): It uses a multi-task training pipeline to align speech, text and semantics through contrastive and generative objectives, enabling robust instruction-following and long-form audio dialogue understanding.
- VITA-Audio Long et al. (2025): It tackles the latency bottleneck in LSLMs by introducing a fast interleaved decoding mechanism and dynamic token predictor, allowing efficient and streaming-capable audio response generation.
- LLaMA-Omni Fang et al. (2025): It extends a unified language model to support real-time speech understanding and generation by integrating low-latency audio streaming, codec-based tokenization, and tightly aligned speech–text representations for seamless multimodal interaction.
- GLM-4-Voice Zeng et al. (2024): It introduces a unified end-to-end spoken language model that interleaves speech and text modalities using a supervised speech tokenizer and joint training paradigm, enabling high-quality spoken dialogue generation.
- Mini-Omni Xie & Wu (2024a): It enables real-time speech interaction by generating text and audio tokens in parallel within one model, using text-instructed parallel decoding and a lightweight training pipeline to preserve the base model's reasoning ability.

- SLAM-Omni Chen et al. (2025): It enables end-to-end spoken dialogue by modeling text and semantic audio tokens in parallel within a single model, supporting zero-shot timbre control and low-latency voice interaction through single-stage training.

A.9 Training Details

We train our model using the AdamW optimizer with a global batch size of 2048, a learning rate of $2e^{-5}$, and a weight decay factor of $1e^{-2}$. The learning rate follows a cosine decay schedule with a linear warmup ratio of 0.01. Training incorporates three stochastic strategies: (1) batchwise AR & NAR objective mixing with probability 0.3; (2) prefix preservation masking with ratio 0.3; (3) stochastic span truncation with probability 0.5. During inference, the model alternates between AR text decoding and NAR diffusion-based audio generation, where text decoding uses nucleus sampling with $k = 10$ and $p = 0.95$, and audio spans are generated with 200 diffusion steps, a block length of 32 tokens, and a total diffusion span length of 640 tokens under classifier-free guidance with scale 0.1. Since different training strategies lead to varying convergence speeds, reported results are based on checkpoints where training loss has converged. All experiments are conducted on 4 nodes with 8 NVIDIA A100 GPUs per node using the DeepSpeed runtime.

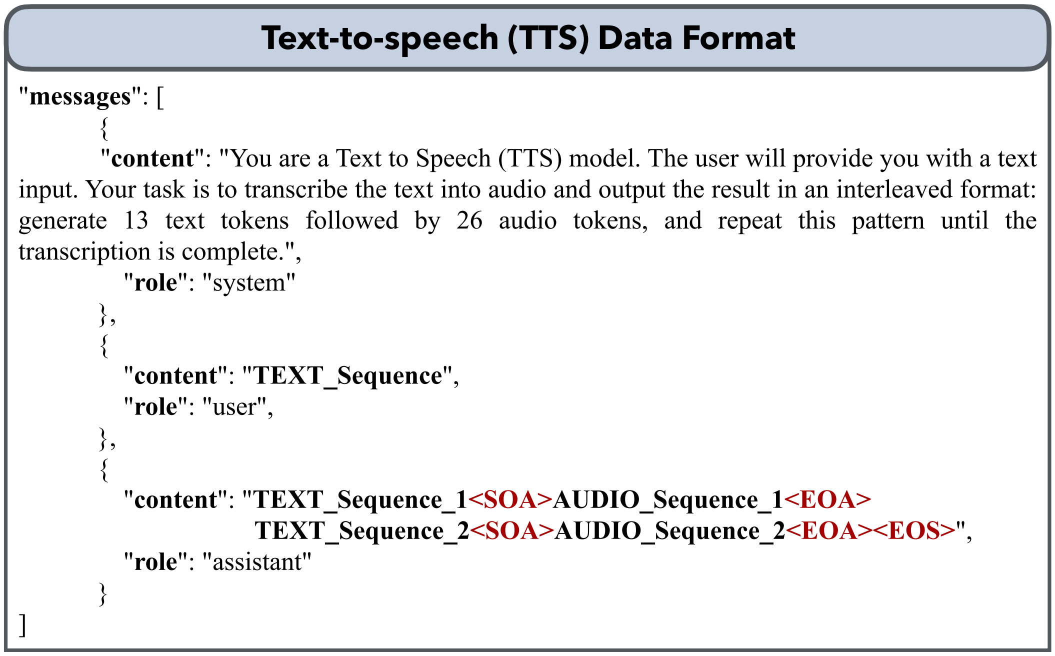
A.9.1 Data Format of Training Data

To enable unified training across diverse tasks, we transform all datasets into a consistent input–output format. On the one hand, this standardization allows the model to seamlessly integrate heterogeneous modalities such as speech, text, and interleaved audio–text sequences. On the other hand, a unified design is essential for supporting our training strategies, including batchwise AR & NAR objective mixing, prefix preservation masking, and stochastic span truncation. These strategies rely on a shared representation to operate across modalities in a consistent way. For clarity, we provide representative examples of the adopted data formats as follow, covering ASR, TTS, audio chat, text chat, AAC, SEC, ASC, and interleaved text–audio data.

Automatic Speech Recognization (ASR) Data Format

```
"messages": [
        {
            "content": "You are an Automatic Speech Recognition (ASR) model. The user will provide you with an audio input. Your task is to transcribe the audio into text and output the result in an interleaved format: generate 13 text tokens followed by 26 audio tokens, and repeat this pattern until the transcription is complete.",
          "role": "system"
        },
        {
          “content": "<SOA>AUDIO_Sequence<EOA>",
          "role": "user",
        },
        {
          "content": "TEXT_Sequence_1<SOA>AUDIO_Sequence_1<EOA>
                      TEXT_Sequence_2<SOA>AUDIO_Sequence_2<EOA><EOS>",
          "role": "assistant"
        }
]
```

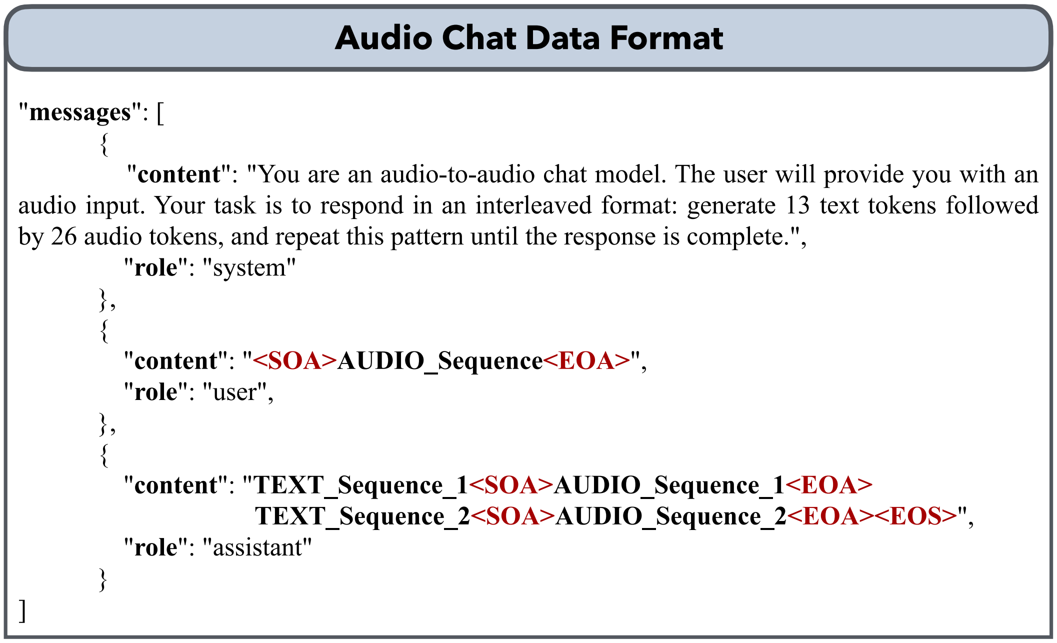
Figure 4: Example of ASR data format.



Text-to-speech (TTS) Data Format

```
"messages": [
          {
          "content": "You are a Text to Speech (TTS) model. The user will provide you with a text input. Your task is to transcribe the text into audio and output the result in an interleaved format: generate 13 text tokens followed by 26 audio tokens, and repeat this pattern until the transcription is complete.",
             "role": "system"
          },
          {
             "content": "TEXT_Sequence",
             "role": "user",
          },
          {
             "content": "TEXT_Sequence_1<SOA>AUDIO_Sequence_1<EOA>
                        TEXT_Sequence_2<SOA>AUDIO_Sequence_2<EOA><EOS>",
             "role": "assistant"
          }
]
```

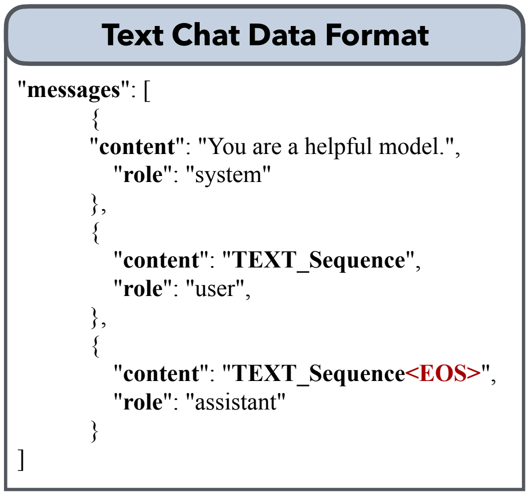
Figure 5: Example of TTS data format.



Audio Chat Data Format

```
"messages": [
          {
             "content": "You are an audio-to-audio chat model. The user will provide you with an audio input. Your task is to respond in an interleaved format: generate 13 text tokens followed by 26 audio tokens, and repeat this pattern until the response is complete.",
             "role": "system"
          },
          {
             "content": "<SOA>AUDIO_Sequence<EOA>",
             "role": "user",
          },
          {
             "content": "TEXT_Sequence_1<SOA>AUDIO_Sequence_1<EOA>
                        TEXT_Sequence_2<SOA>AUDIO_Sequence_2<EOA><EOS>",
             "role": "assistant"
          }
]
```

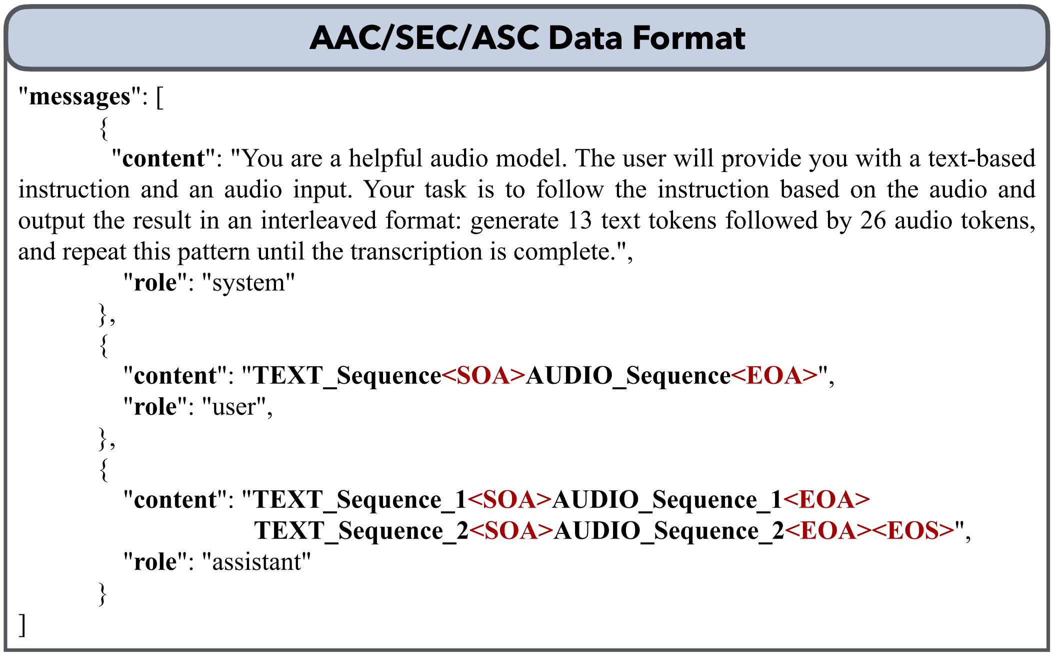
Figure 6: Example of audio chat data format.



Text Chat Data Format

```
"messages": [
        {
        "content": "You are a helpful model.",
          "role": "system"
        },
        {
          "content": "TEXT_Sequence",
          "role": "user",
        },
        {
          "content": "TEXT_Sequence<EOS>",
          "role": "assistant"
        }
]
```

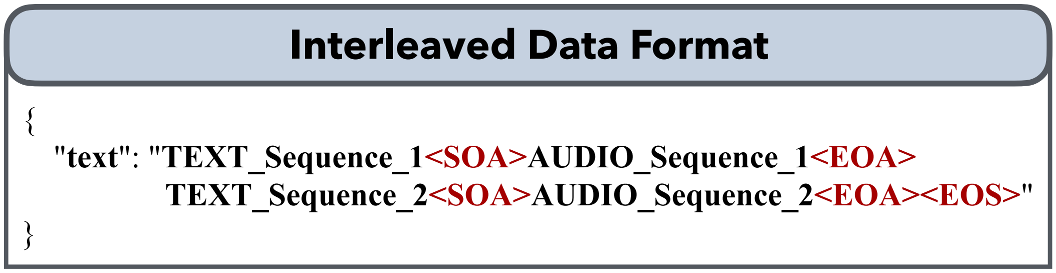
Figure 7: Example of text chat data format.



AAC/SEC/ASC Data Format

```
"messages": [
        {
          "content": "You are a helpful audio model. The user will provide you with a text-based instruction and an audio input. Your task is to follow the instruction based on the audio and output the result in an interleaved format: generate 13 text tokens followed by 26 audio tokens, and repeat this pattern until the transcription is complete.",
          "role": "system"
        },
        {
          "content": "TEXT_Sequence<SOA>AUDIO_Sequence<EOA>",
          "role": "user",
        },
        {
          "content": "TEXT_Sequence_1<SOA>AUDIO_Sequence_1<EOA>
                      TEXT_Sequence_2<SOA>AUDIO_Sequence_2<EOA><EOS>",
          "role": "assistant"
        }
]
```

Figure 8: Example of AAC/SEC/ASC data format.



Interleaved Data Format

```
{
  "text": "TEXT_Sequence_1<SOA>AUDIO_Sequence_1<EOA>
           TEXT_Sequence_2<SOA>AUDIO_Sequence_2<EOA><EOS>"
}
```

Figure 9: Example of interleaved data format.

A.10 Usage of LLM

In this paper, the LLM is employed solely for text refinement—correcting typos, fixing spelling errors, and enhancing readability. It is not used for generating research ideas, producing results, or creating original content.